BANDWIDTH-EFFICIENT ENCODER FRAMEWORK FOR H.264/AVC SCALABLE EXTENSION

Yi-Hau Chen, Tzu-Der Chuang, Yu-Jen Chen, and Liang-Gee Chen

DSP/IC Design Lab.,
Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan
Email: {ttchen, peterchuang, yjchen, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

Scalable video coding is the state-of-art video coding standard for the future streaming applications over heterogeneous networks. Although SVC is based on H.264/AVC, some new coding tools increase the its difficulty in encoder hardware design, especially in memory issues. In this paper, hierarchical B-frame structure and fine granularity scalability are introduced and their memory bandwidth overhead are analyzed. Two method, adaptive spatial-temporal hierarchical ME and scan bucket algorithm are proposed to reduced bandwidth overhead 55% and 88%, respectively, while these algorithms have almost no quality degradation. The concept of proposed encoder framework can be further applied in high definition SVC encoder designs.

1. INTRODUCTION

In the past years, video coding efficiency is always the main target of traditional video coding standards, such as MPEG-2 and H.264/AVC. To meet the requirements from prevalent streaming multimedia applications over heterogeneous network, the ISO/IEC and ITU-T VCEG from the Joint Video Team (JVT) start to call for proposals of scalable video coding (SVC). These proposals are merged into the a Joint Scalable Video Model (JSVM) as the scalable extension of H.264/AVC [1][2]. Currently, the scalable baseline profile is finalized in July 2007, and the other profiles will be finalized soon. The SVC provides three main types of scalability, temporal scalability, spatial scalability, and SNR scalability so that SVC can be used for multi-resolution, frame-rate adaptation, or bandwidth-fluctuated transmission applications. It means that various network streaming applications such as mobile phone, personal computer, and high-definition TV (HDTV) can display the same video with different specifications from one scalable-encoded bitstream.

Figure 1 illustrates the encoder structure of SVC encoder with 3 spatial layers. In SVC, temporal scalability is achieved by hierarchical B-frame coding structure [3] instead of well-known "IBBP" prediction scheme in MPEG-2/4; spatial scalability is based on pyramid coding scheme and many adaptive inter-layer prediction techniques are adopted to exploit the inter-layer correlations between layers; SNR scalability is realized by adding enhancement layer information and embedded bit-plane coding techniques. SVC also allows on-the-fly bit-stream adaptation in these three dimensional scalabilities according to the receiver applications and network conditions.

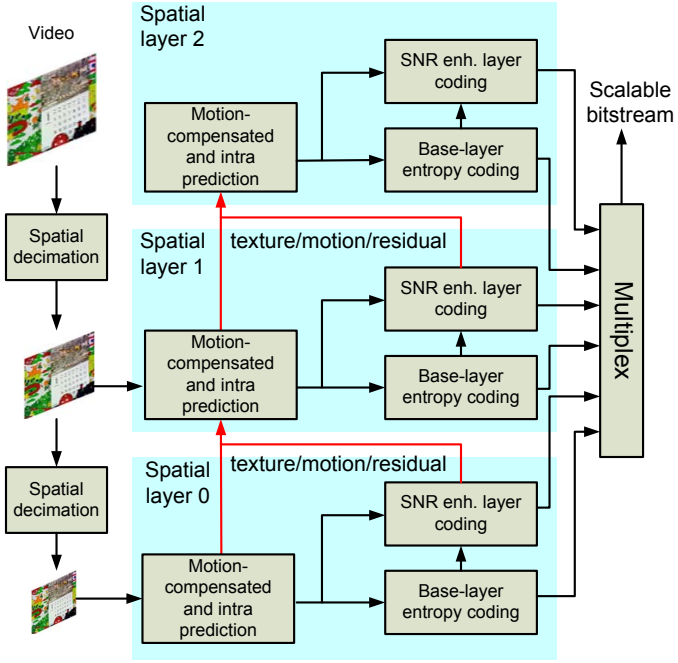


Fig. 1. SVC encoder structure with three spatial layers.

In our previous work[4, 5], the VLSI architecture of multi-level $5/3$ and $1/3$ MCTF coding scheme has been analyzed and memory issue is regarded as an important design challenges in SVC hardware design. In this paper, we will consider the whole SVC encoder which also includes fine granularity scalability (FGS) and propose a bandwidth-efficient SVC encoder framework. The differences of hardware be-

tween H.264/AVC and SVC are discussed and the new design challenges are addressed. In temporal prediction engine, an adaptive spatial-temporal hierarchical motion estimation is adopted by utilizing the information between spatial layers and hierarchical B-frame scheme. In FGS, a scan bucket algorithm is proposed to solve huge external memory bandwidth from frame-level data accessing. Then a case study is given to evaluate the performance of proposed bandwidth-efficient SVC encoder framework.

This paper is organized as follows. In Section 2, the three main scalabilities of SVC and their corresponding techniques are analyzed. Then, the design challenges of an SVC encoder and the proposed algorithms in SVC encoder framework are given in section 3 and 4, respectively. A case study is addressed in section 5. Section 6 will conclude this paper.

2. SCALABLE VIDEO ENCODER STRUCTURE

In this section, we will introduce the new-adopted coding techniques, hierarchical B-frame, inter-layer prediction, and FGS, for three main scalabilities in SVC, respectively.

2.1. Hierarchical B-frame in Temporal Scalability

Temporal scalability allows single bitstream to be decoded and displayed at multiple frame rates. In previous coding standards, only very few choices of frame rate can be supported. In SVC, multi-level hierarchical B-frame [3] is utilized to support temporal scalability so that more decoded frame rate can be provided. As shown in Fig. 2, a 3-level hierarchical B-frame coding structure is composed of 8 consecutive frames. Since hierarchical B-frame can exploit more temporal correlations among one group-of-pictures (GOP), the coding efficiency can be improved with efficient frame-level bit allocation strategy [3].

However, in hierarchical B-frame structure, the long temporal distance between lower temporal levels' frames implies the difficulty in temporal prediction. For example, in Fig. 2, the movement of the object between two key frames could be eight times than that of two neighboring B frames. That is, in key frames' motion estimation (ME), the searching range should be enlarged in order to keep the efficiency of temporal prediction. As described in [6], in video encoder hardware design, the searching range of ME directly influences the requirements of external memory bandwidth and internal memory size if the full-search ME is applied.

2.2. Inter-layer Prediction in Spatial Scalability

Spatial scalability in SVC is realized by decimating the original video sequences into a set of pyramid videos as shown in Fig. 1. In each spatial layer, the original or decimated frames are predicted and encoded as the same with H.264/AVC. To improve the coding efficiency of spatial scalability mode, the

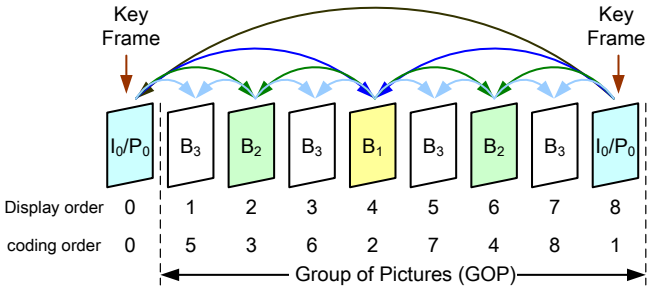


Fig. 2. Closed-loop hierarchical B-frame coding scheme in SVC

redundancy information among spatial layers are removed by the adaptive inter-layer prediction techniques : intra texture, motion and residual prediction. These data are upasmpled from base layer (decimated frames) according to the scaling ratio between layers after the smaller frames are encoded.

2.3. FGS in SNR Scalability

In SVC, SNR scalability can be provided via two main strategies, coarse grain scalability (CGS) and fine grain scalability (FGS). However, CGS can only provide several pre-defined quality points. On the other hand, FGS is aimed arbitrarily truncate the original bitstream to provide more flexibility while maintaining good coding performance throughput whole frame.

Figure 3 shows the block diagram of FGS encoding in case of three enhancement layers. First, four SNR layers are generated from four cascaded reconstruction loops with different QPs. The QP step size is six between adjacent layers. In each enhancement layer, only the differences between transformed coefficients and accumulated inverse-quantized ones from the preceding layers are coded.

According to the coefficients in preceding layers, the coefficients in enhancement layers are classified into two categories, i.e. new coefficients (*NC*s) and refinement coefficients (*RC*s). *NC* is a coefficient which is never significant in all preceding layers. The whole value of *NC* should be coded; *RC* is a coefficient which is significant in any preceding layers. The value of *RC* is limited in {-1, 0, +1}. In Fig. 3, the quantized coefficients in enhancement layers are classified into *NC*s and *RC*s, and then the values of *RC*s are truncated within -1 to 1.

To achieve progressive improvement on entire frame, FGS coding order applies multiple scans through every macroblocks (MBs) in one frame, instead of the traditional MB-by-MB order. As shown in Fig. 3, the coefficients in entire frame are stored in frame memory, and the FGS scan and entropy coding of enhancement layers are frame-level operations. Fig. 4 shows the FGS coding order in JSVM 7 with only four blocks in one frame and eight coefficients in one block for simplicity. In every scan, all blocks in the whole frame are examined in

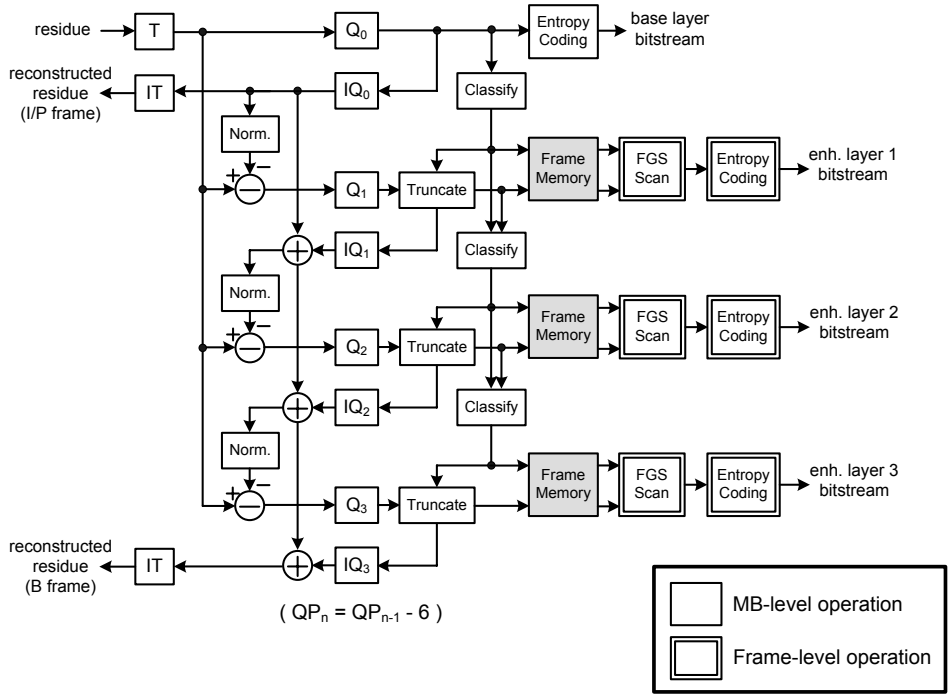


Fig. 3. Block diagram of FGS encoding in H.264/AVC scalable extension

	Coeff. in zigzag scan							
Block 0	0	0	0	1	0	A	0	0
Block 1	B	0	1	C	0	0	0	0
Block 2	1	1	D	0	0	0	0	0
Block 3	0	E	F	0	G	1	0	0

0 : NC = 0 (insignificant NC)
1 : NC ≠ 0 (significant NC)
A~G : RC

(a) Coefficients

	Block 0	Block 1	Block 2	Block 3
Scan 0	0,0,0,1	B	1	0,0,1
Scan 1		0,1	1	E
Scan 2			D	F
Scan 3		C	NC end	
Scan 4	NC end	NC end		G
Scan 5	A			
Scan 6				NC end

(b) Coding order

Fig. 4. Example of FGS scan

turn. In the *k*-th scan, the *k*-th coefficient in zigzag scan is first considered. There are two conditions. If the *k*-th coefficient is *NC*, all *NC*s before next significant one are coded, but *RC*s in the path are skipped. If the *k*-th coefficient is *RC*, only this coefficient is coded. As a result, the coding order in Fig. 4 (b) is *{0,0,0,1}*, *B*, *1*, *{0,0,1}*, *{0,1}*, *1*, *E*, *D*, ..., *G*, *A*, *NC end*.

3. EXTERNAL MEMORY BANDWIDTH OVERHEAD ANALYSIS

The main external bandwidth (BW) overhead of SVC compared to previous H.264/AVC encoder can be compiled into two main categories. The first is from the larger searching range data requirement of ME. The another is the frame-level data accessing of FGS.

In previous video encoder designs [7], Level C data reuse scheme is applied to save the external memory bandwidth of searching range data. If the searching range (SR) is Horizontal : $[-SR_H, SR_H)$ and Vertical : $[-SR_V, SR_V)$, the current block (CB) of size $N \times N$ and the corresponding search region are as shown in Fig. 5. The required SR memory size and external BW for one B-frame can be roughly modeled as follows :

$$Mem.size = (2N + 2SR_H) \times (2SR_v + N) \times 2 \quad (1)$$

$$BW = N \times (2SR_v + N) \times 2 \times (\#\ of\ MB) \times (frame\ rate). \quad (2)$$

For a 4CIF 30Hz video sequences with searching range $SR_H = 64$ and $SR_V = 64$, about 46 KBytes memory size and 219 MByte/sec bandwidth are required. However, if temporal distance of key frames is taken into consideration, for a 4-level hierarchical B-frame structure composed of 16 frames, the SR_H and SR_V may be enlarged according to temporal distance and induce higher memory requirements.

For FGS, in the case of 4CIF 30Hz with three enhancement layers, direct implementation of the three frame memories in Fig. 3 takes $3 \times 704 \times 576 \times 13bits \div 8bits \times 1.5(luma\ and\ chroma) = 2.97MBytes$ while the coefficient length is assumed to be 13 bits. They are too huge to be implemented as on-chip SRAMs. Therefore, storing these enhancement layer coefficients in external DRAM for accessing is a must. However, the frame-level operation of FGS scan needs to access external memory up to 178 MBytes/sec (2.97 MBytes $\times$ 30 frames per second $\times$ read/write).

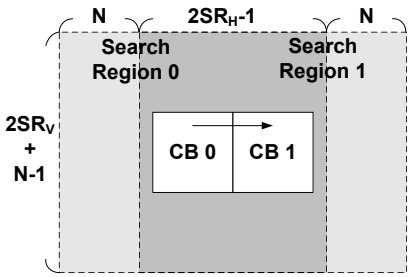


Fig. 5. The current block (CB) and search region for block matching algorithm motion estimation. SR_H and SR_V represent the horizontal and vertical searching range, respectively.

Note that these BW number will be even higher if more than one spatial layer is considered. Therefore, it is necessary to reduce the large external memory bandwidth so that the system bus congestion can be relieved, and the huge power consumption from external memory access can be avoided, too.

4. PROPOSED HARDWARE-ORIENTED BANDWIDTH-SAVING ALGORITHM

4.1. Adaptive Spatial-Temporal Hierarchical ME

To alleviate the large external BW and internal SR memory size from hierarchical B-frame, we adopt the concept of hierarchical ME into SVC encoding structure. As shown in Fig. 6, in smaller frames, the full-search block matching algorithm (FSBMA) is applied and a sufficient SR is assigned to insure the best coding efficiency. For larger frames such as 4CIF or HD resolution, since SVC utilize pyramid structure to achieve spatial scalability and the smaller frames must be encoded first, the base-layer motion vectors are upsampled to predict the possible locations of current motion vectors.

First, the upsampled motion vectors of the MBs in the same row will be gathered to find the most possible vertical range and we perform the Level C data reuse in this region as centric moving row buffer (CMRB). Note that the vertical length of CMRB is much narrow compared to maximal vertical motion vector range so that the external BW can be reduced more. To keep the coding efficiency for sequences of larger or diverse motion, once the upsampled motion vector predictors are out of CMRB, the small yellow region in Fig. 6 will be loaded into SR memory for motion refinement. In Fig. 6, the refinement region is composed of 32×32 pixels due to ± 4 refinements and 6-tap interpolation filter in fractional ME.

Besides, to overcome the larger SR requirement of key frames without much memory size overhead, we combined the two SR memory of B-frame into one larger SR memory for uni-directional ME between key frames. As shown in Fig. 7, the SR of key frame can be enlarged 1.5 times in both

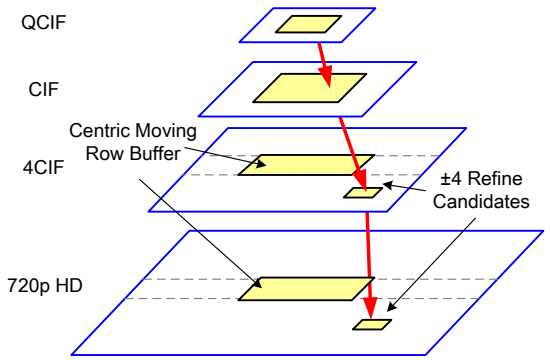


Fig. 6. The concept of adaptive spatial-temporal hierarchical ME scheme. The blue lines depict the whole frame; the yellow regions are the loaded searching range regions for ME; the red line represents the derived motion vectors.

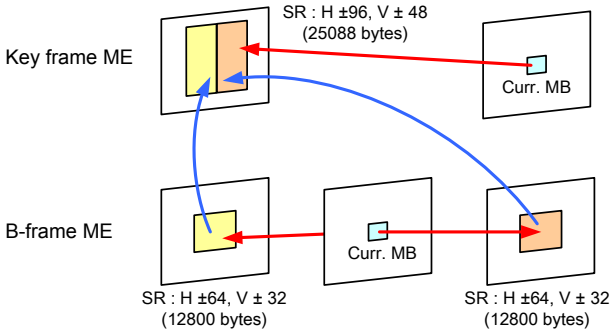


Fig. 7. The concept of reconfig 2 B-frame SR memory to enlarge key frame's SR.

horizontal and vertical directions compared to B-frames. By above two techniques, the proposed adaptive spatial-temporal hierarchical ME can efficiently reduce the external memory BW while limiting the internal SR memory size.

4.2. Scan Bucket Algorithm for FGS

The main concept of proposed scan bucket algorithm is to move FGS scan in Fig. 3 from frame level to MB level. Although the coding order is composed of many scans through all blocks in the whole frame, it can be modified to a more convenient way. At first, we decide the which coefficients are coded in each scan of one block. Then, the partial processed data are stored in internal memory, and then the external memory is exploited as transpose memory to conform to the correct coding order. Figure 8 shows the concept of this method following the example in Fig. 4. When block 0 is processed in MB level, the coefficients are analyzed based on the scan rule, and then put $\{0,0,0,1\}$ into bucket 0, *NC end* into bucket 4, and *A* into bucket 5. Then, block 1 to 3 are processed in turn. Once the bucket is full, the data in that bucket are trans-

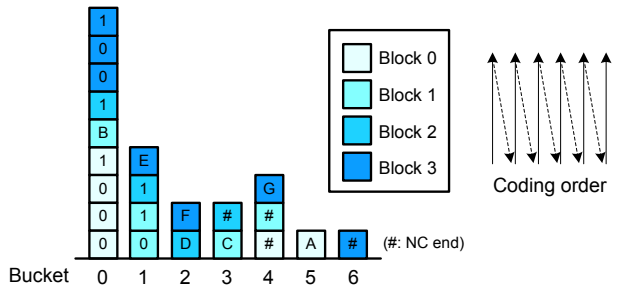


Fig. 8. Scan bucket algorithm corresponding to Fig. 4

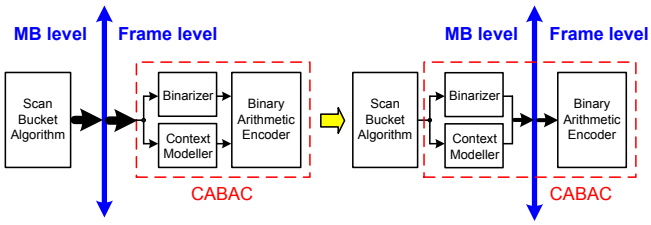


Fig. 9. Concept of early context modeling

ferred to external memory. After all blocks are processed, the external memory is accessed according to scan-by-scan coding order, as shown in Fig. 8. Since the data are well arranged in MB level, external data access becomes regular and simple.

To further improve the external BW reduction, we propose the early context modeling as shown in Fig. 9. The binarizer and context modeller are moved from frame level to MB level. In entropy coding of FGS enhancement layers, context modeling of residual headers requires the information of adjacent (above and left) MBs, which is difficult to retrieve in frame level. This cumbersome problem can be solved by early context modeling in MB level, where relative positions between MBs are explicit. Please note that the early context modeling technique maintains the same context of entropy coding and has no quality degradation.

5. CASE STUDY

In this section, for simplicity, we will discuss the performance of adaptive spatial-temporal hierarchical ME and scan bucket algorithm, respectively. First, we take a SVC structure of 3 spatial layers as example to evaluate the BW reduction performance of adaptive spatial-temporal hierarchical ME. Table 1 shows the specification of searching range of the SVC encoder in this case study. The conventional method is level C data reuse scheme for FSBMA in section 3.

Table 2 shows the comparisons of internal SR memory size requirement and external memory BW between Level C [6] and proposed adaptive spatial-temporal hierarchical ME with centric moving row buffer. The number in Table 2 can

Table 1. Specification of the SVC encoder with 3 spatial layer, GOP=16 frames

Format		QCIF	CIF	4CIF
Level 0	SR_H	48	96	192
	SR_V	24	48	96†
Level 1	SR_H	32	64	128
	SR_V	16	32	64†
Level 2 $\sim$ 4	SR_H	16	32	64
	SR_V	16	32	64†

† : The SR_V represents the maximal motion vector range.
By centric moving row buffer, the SR_V in proposed method is only ± 16.

Table 2. Comparison of memory requirements

		Level C	Proposed
Max. SR mem. size	QCIF	9.2	9.2
(KBytes)	CIF	25.6	25.6
	4CIF	86.5	29.7†
External mem. BW	QCIF	4.47	4.47
(MBytes/sec)	CIF	29.84	29.84
	4CIF	215.17	78.55†
	Total	249.48	112.86†

† : The 32×32 refinement SR regions are included.
The number of loaded refinement blocks are derived from simulation.

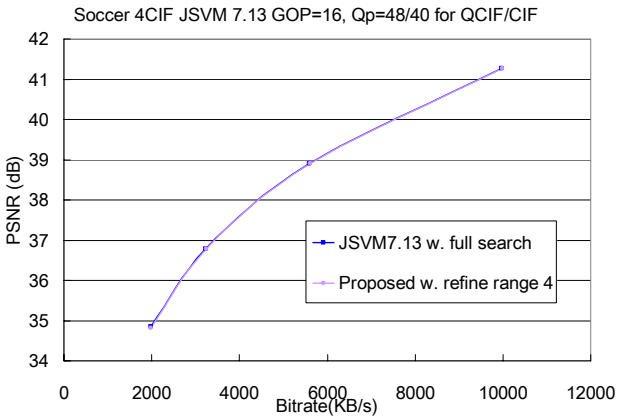
be easily derived from the equations in section 3. Please

The maximal required memory size and external memory BW for two methods are the same in QCIF and CIF format. But in 4CIF format, the centric moving row buffer limits the vertical SR and applies 32×32 refinement block to maintain coding efficiency. As shown in Table 2, the required memory size of proposed method for 4CIF is only 34% compared to Level C scheme and quite similar to the requirement of CIF format. On the other hand, the external memory BW can be also largely reduced 63% in 4CIF and 55% for whole 3 spatial layers. Besides, as shown in Fig. 10, the proposed method has almost the same quality compared to FSBMA ME.

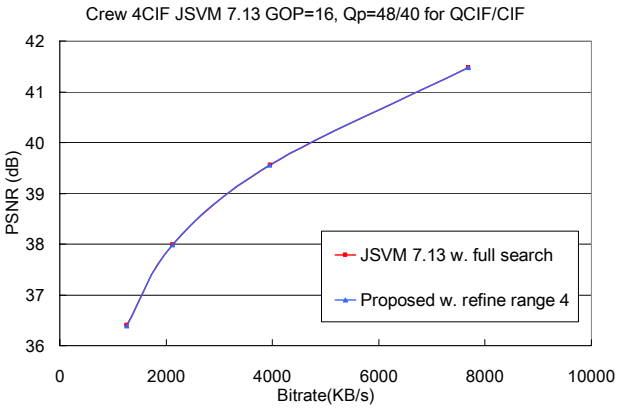
Figure 11 shows the reduction of external memory bandwidth from simulation of 4CIF 704×576 30Hz sequences with GOP=16 frames and only one spatial layer. The result of direct implementation in Fig. 11 can be derived in Sec. 3. The results of proposed method are simulated by verilog-XL and averaged by many standard sequences. The number beside the bar is the reduced percentage out of its above bar. Totally, at most 88% bandwidth reduction is attained when scan bucket algorithm and early context modeling are both used.

6. CONCLUSION

A bandwidth-efficient encoder VLSI architecture framework for H.264/AVC scalable extension is presented in this paper.



(a)Soccer



(b)Crew

Fig. 10. PSNR comparison of proposed adaptive spatial-temporal hierarchical ME and full search algorithm on (a)"Soccer" (b)"Crew" 4CIF 30fps with GOP=16 by JSVM 7.13.

The main differences of SVC from non-scalable H.264/AVC in encoder hardware design are analyzed. To reduce the memory issues induced by hierarchical B-frames and FGS, two main hardware-oriented algorithms, adaptive spatial-temporal hierarchical ME and scan bucket algorithm, are introduced. The former one utilized the inter-layer information of SVC pyramid structure and the proposed centric moving row buffer to limit the internal SR memory size and efficiently reduce the external BW about 55%. The scan bucket algorithm can avoid the irregular and frame-level memory access of FGS and changes it into regular MB-level access to reduce external memory BW about 88% compared to direct implementation. The concept of proposed bandwidth-efficient encoder framework can be further extended to other high definition SVC encoder designs in the future.

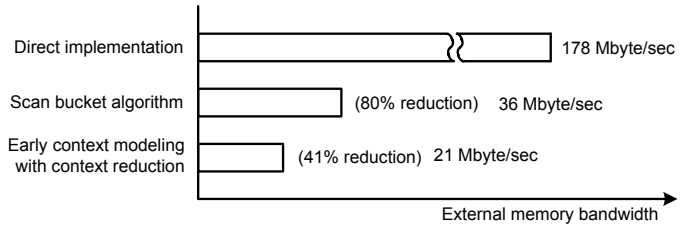


Fig. 11. External memory bandwidth reduction of proposed FGS scan bucket algorithm with the specification of 4CIF 30Hz sequence.

7. REFERENCES

[1] ISO/IEC JTC1, "Joint Draft 8 of SVC Amendment," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-U201, Oct. 2006.

[2] ISO/IEC JTC1, "Joint Scalable Video Model 8.0," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-U202, Oct. 2006.

[3] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE International Conference on Multimedia and Expo*, 2006, pp. 1929–1932.

[4] C.-T. Huang, C.-Y. Chen, Y.-H. Chen, and L.-G. Chen, "Memory analysis of VLSI architecture for 5/3 and 1/3 motion-compensated temporal filtering," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[5] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, C.-J. Lian, and L.-G. Chen, "System analysis of VLSI architecture for motion-compensated temporal filtering," in *Proc. IEEE International Conference on Image Processing*, 2005.

[6] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. CSVT*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[7] Y.-W. Huang and et al., "A 1.3tops H.264/AVC single-chip encoder for HDTV applications," in *Proc. of IEEE ISSCC*, 2005, pp. 128–588.